



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/960,529	09/21/2001	Benjamin Renaud	BEAS-01067US0	5297
23910	7590	09/22/2004	EXAMINER	
FLIESLER MEYER, LLP FOUR EMBARCADERO CENTER SUITE 400 SAN FRANCISCO, CA 94111			VU, TUAN A	
			ART UNIT	PAPER NUMBER
			2124	

DATE MAILED: 09/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/960,529

Applicant(s)

RENAUD, BENJAMIN

Examiner

Tuan A Vu

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 21 September 2001.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-54 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-54 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 21 September 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☒ Notice of References Cited (PTO-892) ⁶
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 20030311 ⁶
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application (PTO-152)
- ☐ Other: _____

DETAILED ACTION

1. This action is responsive to the application filed September 21, 2001.

Claims 1-54 have been submitted for examination.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knutson, USPN: 6,557,100 (hereinafter Knutson), in view of Seidman et al., USPubN: 2003/0005166 (hereinafter Seidman).

As per claim 1, Knutson discloses a method of automatically deploying an application across a distributed computing domain including a plurality of processing devices, the method comprising:

automatically scanning for an undeployed application stored in an application archive file accessible to at least one of the plurality of processing devices (e.g. Fig. 6; Fig. 7 – Note: identification of applications that need to be redeployed is equivalent to scanning for undeployed application, accessible to at least one of the client machines);

recognizing an undeployed application in the application archive file (e.g. Fig. 6,7 – Note: creation of new file is equivalent to deploying to application being absent or never deployed from the previously deployed files); and

deploying the undeployed application to a portion of the plurality of processing devices, such that the application is capable of being executed by the portion of the plurality of processing devices (Note: enterprise Java beans being redeployed in JAR package implicitly discloses application being deployed and executed from one environment to another, i.e. by portion of processing devices bound by same enterprise or business-related yet protocol-neutral portability form – see *scalability* - col. 1, line 33 to col. 2, line 12).

Knutson does not explicitly disclose that deploying of the undeployed application is deploying to selected portion of the plurality of processing devices. But in view of the distributed aspect of the EJB within a LAN and implementation of session-oriented of such distribution by Knutson (see col. 3, lines 32-67; *security rules, Session beans* – col. 4, line 42 to col. 5, line 3), the portion of clients permitted via Lan-based rules and session authentication is equivalent to client machines being selected among the plurality of LAN processing devices. Hence, Knutson implicitly discloses deploying to selected portion of plurality of processing devices.

Nor does Knutson disclose that the application archive file structure is an application directory. But Knutson discloses a structural organization to store the archive file with hierarchy description (Fig. 4-5; *manifest file* – col. 4, lines 44-48) enabling the identification of undeployed application in a service environment where beans deployment program or utilities, e.g. home container or home interface, take place (e.g. col. 2, lines 40-48; col. 5, lines 49-53; *cache Jar 735* – Fig. 7); hence has suggested a implicitly disclosed a file hierarchy or directory-like system operable to depict or store the archive file contents prior to file processing. The technique of setting a server deployment environment with a directory for developing or processing a file as

Art Unit: 2124

taught by Knutson was a known concept in the server deployment technology and is further evidenced by Seidman. In a method to deploy a bean application similar to Knutson, Seidman discloses a bean deploying system where the JAR files are stored in directory particularly associated with bean identification/name (e.g. pg. 7, para 0106-0108). It would have been obvious for one skill in the art at the time the invention was made, in case the JAR files by Knutson is not deployed from an application directory, to implement the directory storing as taught by Seidman because this way bean resources are partitioned according to specific and related information and that would make it easier to track down, reuse or re-deploy resources according to Seidman, and also benefit from the commonly known practice of organizing computer files which are susceptible to be open and processed in directories inside a computer system as suggested by Knutson's JAR processing.

As per claim 2, Knutson discloses the steps of:

obtaining a list of applications stored in the application directory (e.g. *cached* -col. 2, line 40-45 – Note: descriptor entries being organized in cache are equivalent to list);

comparing the list of applications stored in the application directory to a list of previously deployed applications in order to select the application to be deployed, and deploying the selected application to the selected portion of the plurality of processing devices (e.g. col. 5, lines 42-51; Fig. 7 – Note: comparing with previously deployed descriptor being cached is equivalent to comparing against list of application stored in application directory).

As per claim 3, see Knutson (Fig. 7; col. 5 line 42 to col. 6, line 5 – Note: for any change identified from comparing with cached list, a new version of file is created, and this reads

Art Unit: 2124

on an application being absent from the previously deployed list because a newer version is not in the currently deployed list)

As per claims 4 and 5, see Knutson (e.g. *descriptor* - Fig. 7; col. 5 line 42 to col. 6, line 5 - Note: descriptor detected from parsing a archive structure is equivalent to attribute of a file containing bean component)

As per claim 6, Knutson does not explicitly discloses a file date as attribute; but official notice is taken that versioning of a file with incorporating a date attribute therein was a well known concept at the time the invention was made. In view of Knutson updating of a version (col. 5, lines 26-34), the limitation to making a attribute date is implicitly disclosed or would have been obvious because incorporating a date as attribute for versioning file enables clear distinguishing of versions using a time base, a concept universally known as non-repetitive or un-duplicable.

As per claim 7, Knutson discloses indicator being attribute of a file associated with a file containing the application (Fig. 7 - Note: EJB descriptor is attribute of bean contained in JAR file)

As per claim 8, in view of the rationale for obviousness regarding the identification a file version by attribute using system date as set forward in claim 6, the setting of a EJB identifier using a date attribute would also have been obvious according to known concept as set forth above.

As per claim 9, Knutson discloses analysis of attributes from undeployed application (e.g. step 725 - Fig. 7- Note: descriptor identified as not being cached or different from a previously cached descriptor is attribute of undeployed application) and attributes from

Art Unit: 2124

distributed computing domain (cached descriptor or JAR- step 735 Fig. 7 – Note: redeploying of application to a selected portion of clients has been disclosed in view of rationale in claim 1).

As per claim 10, by virtue of the rationale in claim 6, the limitation as to incorporating a date attribute in a descriptor or deployment descriptor as mentioned therein would also have been obvious according such rationale.

As per claim 11, Knutson does not disclose a configuration file but discloses other descriptors in a package to be analyzed by a service computer used for redeployment (Fig. 3). Analogous to the service to identify which application bean to track and deploy beans by Knutson, Seidman discloses a descriptor file being configured as a XML grammar type file enabling a tracking program to identify and generate the bean based on deployment specification via a network computing and file communication (*deployment file, descriptor file* - pg. 3, para 0030-0031; pg. 7, para 0104). Thus, it would have been obvious for one skill in the art at the time the invention was made to provide a XML descriptor file with attributes as taught by Seidman to Knutson's analysis of descriptors. XML meta-file has been used more as an unified meta description format and was a well-supported feature in distributed system at the time the invention was made; and since EJB communication pertains to such web-based paradigm of distribution such as suggested by Seidman (Fig. 3, 4), the service of identifying and tracking of browser data using such increasingly accepted form of meta description or configuration instruction, e.g. as taught by Seidman's descriptor file, would enable more wide-spread fetching or deploying objects or data used in browser applications in that it provides a large-scale and unified approach for better supporting and facilitating browser based applications (see

Art Unit: 2124

Seidman's Background) in managing and supplying bean instances to enterprise clients as intended by Knutson.

As per claim 12, Knutson does not disclose attributes as tags of XML language, but in view of the teachings by Seidman from above, this feature would also have been obvious in light of the benefits as put forth above in claim 11.

As per claim 13, only Seidman discloses a automated schedule for synchronization of data (pg. 2, para.0022; pg. 9, claim 20). This is evidence that enterprise business bounding client machines and server and being equipped with automated program for enforcing synchronizing of application or data between server persistent storage and client local storage was a known concept at the time the invention was made. Hence, since Knutson also provides a form of synchronizing service so to update versions being in use by client applications with a newer version of applications, it would have been obvious for one skill in the art at the time the invention was made to implement the automated update service operable on time interval as suggested by Seidman to Knutson's service, because that way enterprise business data or application program would be constantly in sync with the persistent storage as suggested by the approach by Seidman, such synchronizing enabling more secure or fault-free operating system or application level within the executing resources of the enterprise network devices.

As per claim 14, Knutson discloses parsing a JAR file to identify applications that have not been deployed (e.g. col. 5, lines 42-51), hence has disclosed undeployed application being contained in a single file.

As per claim 15, Knutson discloses beans being separate files (Fig. 3; *class files* - col. 4, lines 44-48 – Note: class identified from a Jar are different files).

Art Unit: 2124

As per claim 16, see Fig. 1B.

As per claim 17, in view of the rationale as set forth in claim 1 concerning providing a application directory, like a file system directory, to store JAR file contents, the non-volatile information limitation would have been obvious for the same rationale therein.

As per claim 18, Knutson discloses a list of previously deployed application accessible to one of processing devices (e.g. step 735 Fig. 7)

As per claim 19, Knutson discloses a method for automatically maintaining an application object across a distributed computing domain, the object contained within one application file, and said computing domain including a plurality of processing devices, the method comprising the steps:

retrieving a list of all of the application files located within an application archive structure (e.g. steps 600-615- Fig. 6; steps 700-715 – Fig. 7);

comparing the list of all of the files located within an application archive structure to a list of all of the files associated with previously deployed application objects(e.g. col. 5, lines 42-62);

for each application file, deploying the application object contained in the application file when the application file is absent from the list of all the files associated with previously deployed application objects (e.g. col. 5, line 47 to col. 6, line 5 – Note: if a new file is to be compiled due to non-matching between lists of files from descriptor comparing, creation of new file is equivalent to deploying to application being absent from the previously deployed files)

Art Unit: 2124

for each application file, redeploying the application object contained in the application file when the application file differs from the corresponding file on the list of all of the files associated with previously deployed application objects (e.g. col. 5, lines 42-67).

But Knutson does not disclose list of application located in an application directory; but this limitation has been addressed in claim 1 above.

Nor does Knutson disclose that for each application file on the list of all of the files associated with previously deployed application objects, undeploying the application object associated with an application file when the application file on the list of all of the files associated with previously deployed application objects is absent from the list of all of the application files located within the application directory. But in view of the creation of new files as a result of descriptor comparison mismatch, the suggestion as to download or transmit the latest compiled application bean to the user also entails the use of the latest compiled bean and activation of such bean at the client processor. Hence, the concept of undeploying an older version at the client machine is suggested. Hence, it would have been obvious for one skill in the art at the time the invention was made to undeploy any application file being previously used at the client end which is unmatched against the application JAR list files so that the resources can be directed to using the new created file being compiled at the server deployment directory.

As per claim 20, Knutson discloses difference from comparing the value of a deployment indicator associated with an application file with the value of a deployment indicator recorded on the list of previously deployed application objects (see Fig. 6,7).

As per claims 21-26, these claims correspond to claims 5-8, 13, 16, respectively; hence are rejected using the corresponding rejection as set forth therein.

Art Unit: 2124

As per claims 27 and 28, see Knutson: Fig. 1A-B.

As per claims 29-30, see corresponding rejections of claims 17-18, respectively.

As per claim 31, Knutson discloses a method of automatically maintaining an application object across a distributed computing domain including a plurality of processing devices, the method comprising:

automatically scanning an application directory to create a list of application objects located in the application archive structure (e.g. Fig. 6; Fig. 7 – Note: identification of applications that need to be redeployed is equivalent to scanning for undeployed application, accessible to at least one of the client machines);

comparing the list of application objects to a list of previously deployed application objects to find a previously deployed application object corresponding to each application object in the application directory (e.g. col. 5, lines 42-51; Fig. 7 – Note: comparing with previously deployed descriptor being cached is equivalent to comparing against list of application stored in application directory); and

generating a response to the step of comparing, wherein the response is deploying the application object to a selected portion of the plurality of processing devices when the application object is absent from the list of previously deployed application objects (col. 5, line 47 to col. 6, line 5);

redeploying the application object to the selected portion of the plurality of processing devices when the application object in the application directory is more recent than the corresponding application object on the list of previously deployed application objects (e.g. col.

Art Unit: 2124

2, lines 39-48; Fig. 6,7 – Note: recreating of new compiled application in response to version mismatch or difference is equivalent to redeploying of application)

But Knutson does not explicitly disclose undeploying the application object when the application object on the list of previously deployed application objects is absent from the list of application objects located in application directory. But this limitation has been addressed in claim 19 above.

As per claims 32-36, these claims correspond to claims 26-30, respectively; hence are rejected using the corresponding rejection as set forth therein.

As per claims 37-39, these are computer medium claims with medium to embody instructions for performing the method claims 1, 19 and 31 respectively, which Knutson also discloses (see Knutson: col. 6, lines 19-33)

As per claim 40, Knutson discloses a processing system including a first processing device, a memory accessible by the first processing device, the processing system comprising:

a group of processor readable instructions stored in the memory device and operating the first processing device to perform a group of steps:

automatically (scanning for an undeployed application) stored in an application archive structure accessible to first processing device,

recognizing (undeployed application) in the application archive structure; and

deploying (undeployed application to a selected portion of the processing system); all of which steps having been addressed in claim 1.

But Knutson does not explicitly teach deploying of the undeployed application to selected portion of the plurality of processing devices; nor does Knutson disclose that the application

Art Unit: 2124

archive file structure is an application directory. But these limitations have been addressed in claim 1 above.

As per claim 41, Knutson discloses the selected portion of the processing system includes the first processing device (server 102 – Fig. 1A; col. 4, lines 49 to col. 5, line 7; col. 5, line 55 to col. 6, line 11 – Note: deployment of beans via recompiling effected by the server discloses selected portion including first processing device, e.g. compiling capabilities of server machine)

As per claim 42, Knutson discloses including a second processing device in communication with the first processing device, wherein the selected portion of the processing system includes the second processing device (e.g. Fig. 6,7 – Note: the JAR scanning and selection of application files to be redeployed– or first processing device– in conjunction or communication with the redeployment process, i.e. a second processing device, itself in the same environment, is equivalent to selected portion including a second processing device).

As per claim 43, in view of claim 42, where the selecting for deployment and the deployment process is executed on the same server machine, Knutson has disclosed the first processing device and the second processing device are located on a first computer.

As per claim 44, Knutson discloses the first processing device is located on a first computer and the second processing device is located on a second computer (computer 108, 110, 112 – Fig. 1A).

As per claim 45, this is a processing system version claim including processor readable instructions stored in the memory device and operating the first processing device to perform a group of steps as recited in method claim 19 including the steps

Art Unit: 2124

retrieving a list of all of the application files;
comparing the list;
for each application file, deploying;
for each application file, redeploying; and
for each application file on the list of all of the files associated
with previously deployed application objects, undeploying;
as recited in claim 19.

Hence this claim is rejected with the corresponding rejection as set forth in claim 19.

As per claims 46-49, these claims correspond to claims 41-44, respectively; hence are rejected using the corresponding rejection as set forth therein.

As per claim 50, this is a computer readable medium claim for a processing system including at least a first processing device and a memory device accessible by the first processing device, the processing system comprising instructions to perform a group of steps corresponding to those in claim 31, those steps including:

automatically scanning;
comparing the list; and
generating a response comprising deploying the application object to a selected portion;
redeploying the application object to the selected portion; or
undeploying the application object; all of which exactly as recited in claim 31.

Hence, this claim is rejected using the corresponding rejections as set forth in claim 31.

As per claims 51-54, these claims correspond to claims 46-49, respectively; hence are rejected using the corresponding rejection as set forth therein.

Art Unit: 2124

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please consult Examiner before using this number)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT

Application/Control Number: 09/960,529

Page 15

Art Unit: 2124

September 16, 2004



ANIL KHATRI
PRIMARY EXAMINER